In the Claims:

Currently

1. (Amended) A mass storage system comprising:

a mass storage device;

a cache memory coupled to the mass storage device, the cache memory being organized in data blocks and having a first data block;

a microprocessor coupled to the mass storage device and the cache memory; and

a controller coupled to the microprocessor and the cache memory, wherein the controller:

receives a data request from a host system;

calculates new cache counter and pointer values when the first requested data block is not contained within the first block of the cache;

initiates an auto-transfer of the requested data that resides in the cache to the host system; and

requests a transfer of the requested data that resides in the mass storage device to the host system, wherein the request of the transfer and the initiation of the auto-transfer occurs substantially concurrently.

2. (Previously presented) The mass storage system of claim 1 further comprising a controller register including:

a counter register containing a value for the number of blocks of data in the cache memory,

a start address register identifying the first block of data in the cache memory; and

a pointer register containing a pointer to the first block of data in the cache memory.

3. (Original) The mass storage system of claim 1 wherein the microprocessor transfers the requested data that resides in the mass storage device to the host system by way of the cache memory.

4. (Original) The mass storage system of claim 1 wherein the microprocessor controls the transfer of requested data that resides in the mass storage device and the controller controls the transfer of requested data that resides in the cache.

5. (Original) The mass storage system of claim 1 wherein the controller includes a general or special purpose processor executing program instructions.

6. (Original) The mass storage system of claim 1 wherein the transfer of requested data that resides in the mass storage device occurs substantially simultaneously with the transfer of data that resides in the cache.

7. (Original) The mass storage system of claim 1 wherein the mass storage system and the host system are integrated into a single unit.

8. (Original) A method of retrieving data from a mass storage system comprising:

receiving a data request from a host system, the data request including a block address for a first block of the requested data and a number of blocks in the request;

if none of the requested data is in a cache memory, initiating a transfer of the requested data from a mass storage device;

if a portion of the requested data is in the cache memory and a portion of the requested data is in the mass storage device, transferring the portion of the requested data from the cache

memory to the host system substantially concurrently with transferring the portion of the requested data from the mass storage devices to the host system;

if all the requested data is in the cache memory, transferring the requested data from the cache memory to the host system;

wherein the steps of transferring the requested data from the cache memory system include calculating a starting location in the cache memory for the transfer, based upon the block address and the number of blocks in the request received from the host system.

9. (Original) The method of claim 8 wherein the data request has a first logical address protocol and the cache memory has a second logical address protocol and including the step of translating between the first and second address protocols.

10. (Withdrawn) A controller in a disk memory system for initiating an auto-transfer of host-requested-data from a cache memory to a host system, wherein an initial buffer-counter indicates a total number of blocks of data in the cache memory, wherein an initial buffer-pointer points to a first block of data in the cache memory, and wherein an initial start-address indicates an address of the first block of data in the cache memory, the controller comprising:

a first-circuit receiving as input a request for host-requested-data, generating an output signal when at least a portion of the host-requested-data is within the cache memory;

a second-circuit receiving as input the output signal of the first-circuit and selectively outputting a new value of the buffer-counter, the buffer-pointer and the start-address in response to the output of the first-circuit; and

an auto-transfer circuit receiving as input the new value for the buffer-counter, the new value for the buffer-pointer, and the new value for the start-address and outputting in response

control signals to auto-transfer the at least a portion of the host-requested-data from the cache memory to the host system.

11. (Withdrawn) The controller of claim 10 wherein the auto-transfer circuit controls the auto-transfer without the intervention of a microprocessor that is within the disk memory system.

12. (Withdrawn) The controller of claim 11 wherein the auto-transfer is initiated when an address of a first block of the host-requested-data is different from the initial start-address.

13. (Withdrawn) The controller of claim 10 further comprising:

a third-circuit operable to generate an output when a portion of the host-requested-data is not in the cache memory; and

a microprocessor interface responsive to the output of the third-circuit for invoking a microprocessor to transfer the portion of the host-requested-data that is not in cache memory from a magnetic disk device concurrent with the auto-transfer of other portions of the host-requested-data from cache memory.

14. (Withdrawn) The controller of claim 10 operable to initiate the auto-transfer in response to a read-command by the host system.

15. (Withdrawn) The controller of claim 10 wherein the first circuit, the second circuit, and the auto transfer means are implemented on an integrated circuit.

16. (Withdrawn) The controller of claim 10 wherein the first circuit, the second circuit, and the auto transfer means are implemented within a host interface.

17. (Withdrawn) The controller of claim 10 further comprising registers that store the initial value of the buffer counter, the initial value of the buffer-pointer, initial value of the start-address.

18. (Withdrawn) The controller of claim 17 further comprising registers operable to store the new value of the buffer-counter, the new value of the buffer-pointer, and the new value of the start-address.

19. (Withdrawn) A controller in a disk memory system for initiating host-requested-data from a cache memory to a host buffer-counter indicates a total number of blocks of an auto-transfer of system, wherein a data in the cache memory, wherein a buffer-pointer points to a first block of data in the cache memory, wherein a start-address indicates an address of a first block of data in the cache memory, wherein a task-file-address indicates an address of a first block of the host-requested-data, wherein a transfer-length indicates a total number of blocks of data in the host-requested-data, the controller comprising:

a first-circuit operational to generate a first-output that equals the task-file-address minus the start-address;

a second-circuit operationally coupled to receive the first-output, the second-circuit being operational to generate a second-output only when a content of the buffer-counter is greater than the first-output;

a third-circuit operationally coupled to receive the first-output and the second-output, the third-circuit being operational to produce a third-output that equals the first-output when the third-circuit receives the second-output and to produce a third-output that equals zero when the third-circuit does not receive the second-output;

a fourth-circuit operationally coupled to receive the third-output, the fourth-circuit being operational to generate a fourth-output that equals the content of the buffer-counter minus the third-output;

a fifth-circuit operationally coupled to receive the third-output, the fifth-circuit being operational to generate a fifth-output that equals the third-output added to the value of the start-address;

a sixth-circuit operationally coupled to receive the third-output, the sixth-circuit being operational to generate a sixth-output that equals the third-output added to the contents of the buffer-pointer; and

auto-transfer means for initiating the auto-transfer of the host-requested-data from the cache memory based upon the fourth-output, the fifth-output, and the sixth-output.

20. (Withdrawn) The controller of claim 19 wherein the auto-transfer is initiated without the intervention of a microprocessor that is within said disk memory system.

21. (Withdrawn) The controller of claim 20 wherein the auto-transfer is initiated when the task-file-address is different from the start-address.

22. (Withdrawn) The controller of claim 19 further comprising:

a seventh-circuit operational to generate a seventh-output when the transfer-length is greater than the fourth-output; and

microprocessor interface means responsive to the seventh-output for invoking a microprocessor to transfer host-requested-data' that is not in the cache memory from a magnetic disk device.

23. (Withdrawn) A method of operating a disk-controller in a disk memory system having a microprocessor and a magnetic disk device, the method initiating an auto-transfer of host-requested-data from a cache memory to a host system, wherein an initial Buffer-Counter indicates a total number of data-blocks in the cache memory, wherein an initial Buffer-Pointer points to a first data-block in the cache memory, wherein an initial Start-Address indicates an address of the first data-block in the cache memory, the method comprising:

generating a first-output when at least a portion of the host-requested-data is within the cache memory;

in response to the first-output, generating a new-value for the Buffer-Counter, a new-value for the Buffer-Pointer, and a new-value for the Start-Address, wherein the new-values are based upon a comparison of the initial Start-Address to an address of a first data-block within the host-requested-data; and

initiating, without invoking operation of the microprocessor, the auto-transfer of the host-requested-data from the cache memory to the host system based on the new value for the Buffer-counter, the new value for the Buffer-Pointer, the new value for the Start-Address.

24. (Withdrawn) The method of claim 23 wherein auto-transfer is initiated when an address of a first data-block of the host-requested-data is different from the initial value of the Start-Address.

25. (Withdrawn) The method of claim 23 further comprising:

generating a second-output when none of the host-requested-data is in the cache memory; and

responding to the second-output and invoking operation of the microprocessor to transfer the host-requested-data from the magnetic disk device.

26. (Withdrawn) The method of claim 25 further comprising:

generating a third-output if a first-portion of the host-requested-data is in the cache memory and a second-portion of the host-requested-data is not in the cache memory;

responding to the third-output and initiating auto-transfer of the first-portion of the host-requested data from the cache memory; and

simultaneous with the auto-transfer, invoking operation of the microprocessor to transfer the second-portion of the host-requested-data from the magnetic disk device.

27. (Original) A disk memory system, comprising:

a disk-device for storing data-blocks on disk-storage-media;

a cache for storing data-blocks;

a disk-controller;

registers within said disk-controller containing a cache-start-address of a first data-block in said cache, and a cache-block-length that defines a total number of data-blocks stored in said cache;

said disk-controller receiving a data-request that contains a request-start-address of a first data-block in said data-request, and a request-block-length that defines a total number of data-blocks in said data-request;

a microprocessor operationally interconnecting said disk-device, said cache, and said disk-controller;

logic means in said disk-controller responsive to said cache-start-address as compared to

said request-start-address, and to said cache- block-length-as compared to said request-block-length;

said logic means being operable to determine when no data-blocks corresponding to said data-request reside in said cache, and operating in response to such a determination to cause said microprocessor to fetch said data-blocks corresponding to said data-request from said disk-device;

said logic means being operable to determine when all of the data-blocks corresponding to said data-request reside in said cache, and operating in response to such a determination to cause said disk-controller to auto-transfer all of said data-blocks corresponding to said data-request from said cache without requiring operation of said microprocessor; and

said logic means being operable to determine when a cache-hit-portion of data-blocks corresponding to said data-request reside in said cache and a cache-miss-portion of said data-blocks corresponding to said data-request do not reside in said cache, and operating in response to such a determination to concurrently cause said disk-controller to auto-transfer said cache-hit-portion of said data-blocks corresponding to said data-request from said cache, and to cause said microprocessor to fetch data-blocks corresponding to said cache-miss-portion of said data-request from said disk-device.

28. (Original) A disk memory system, comprising:

a relatively slow disk-device for storing data-blocks on disk-storage-media;

a relatively fast cache for storing data-blocks;

a disk-controller, and a microprocessor;

registers within said disk-controller containing a cache-start-address of a first data-block

in said cache, and a cache-block-length that defines a total number of data-blocks stored in said cache;

said disk-controller receiving as input a data-request from said host-system;

said data request containing a request-start-address of a first data-block in said data-request, and a request-block-length that defines a total number of data-blocks in said data-request;

a logic circuit in said disk-controller responsive to said cache-start address as compared to said request-start-address, and to said cache-block-length as compared to said request-block-length; said logic circuit being operable to determine a cache-miss when no data-blocks corresponding to said data-request reside in said cache, and operating in response to a cache-miss to cause said microprocessor to fetch said data-blocks corresponding to said data-request from said disk-device;

said logic circuit being operable to determine a total-cache-hit when all of the data-blocks corresponding to said data-request reside in said cache, and operating in response to a total-cache-hit to cause said disk-controller to auto-transfer all of said data-blocks corresponding to said data-request from said cache without requiring operation of said microprocessor; and

said logic circuit being operable to determine a partial-cache-hit when a first-portion of data-blocks corresponding to said data-request reside in said cache and a second-portion of said data-blocks corresponding to said data-request do not reside in said cache, and operating in response to a partial-cache-hit to concurrently cause said disk-controller to auto-transfer said first-portion of said data-blocks corresponding to said data-request from said cache, and to cause said microprocessor to fetch data-blocks corresponding to said second-portion of said data-request from said disk-device.

29. (Original) The disk memory system of claim 28 wherein the slow disk-device, the fast cache, the disk controller, and the microprocessor all of which are operationally and electrically interconnected to form a unitary disk memory system that appears as a single source of data-blocks to a host-system.

30. (Original) The disk memory stem of claim 28 wherein the logic means include a processor executing programmed instructions.